EXAMINER'S AMENDMENT

An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.

Authorization for this examiner's amendment was given in a telephone interview with Mr. Aslam Jaffrey,Reg.No. on October 18, 2010.

The application has been amended as follows:

IN THE CLAIMS:

1. (Currently Amended) A method comprising:

executing, by a server device, a multi-threaded application having at least one thread to monitor a port of the server device for received requests from a plurality of client devices;

~~receiving a request from a first client device to multicast a file as a plurality of packets of data from a server device to multiple client devices by executing a multi-threaded application with at least one thread monitoring a port designated for receiving multicast requests;~~

executing, by the server device, at least three request handlers to manage any of the received requests from ~~remote~~ the plurality of client devices, the at least three

request handlers comprising an upload request handler, a multicast download request handler and a unicast download request handler;

detecting, by the server device, that a multicast request is received from a first client device of the plurality of client devices;

invoking, by the server device, the multicast download request handler in order to save session information regarding the received multicast request and create another thread of the multi-threaded application to service the received multicast request;

wherein servicing the received multicast request by the another thread further comprises:

transmitting the plurality of packets of data from **[[**a**]]** the server device to the multiple client devices using a multicast trivial file transfer protocol (TFTP) as a TFTP-compliant flow; and

applying~~, by the server,~~ one or more flow control techniques not defined by the multicast TFTP to the TFTP-compliant flow, wherein the flow control techniques comprise**[[**s**]]** ~~at least~~ determining whether the server device has sufficient resources to satisfy the request based on a block size corresponding to the request and an available bandwidth, and sending an error packet to the first client device if the server device does not have sufficient resources to satisfy the request, wherein applying, by the server, one or more flow control techniques not defined by multicast TFTP comprise delaying a start of the transmission of the plurality of packets, wherein delaying includes if a packet is lost and a send delay is determined to be zero, the send delay of zero is set to one, and if the send

delay is determined to be greater than timeout/four, the send delay is set equal to the timeout/four, wherein the send delay is doubled or the send delay is decreased by one for a number of successful packets received until the send delay reaches zero, wherein the number of successful packets received includes ten.

2. (Cancelled)

3. (Original) The method of claim 1 wherein applying, by the server, one or more flow control techniques not defined by multicast TFTP comprises:

determining whether a request to download the file is a subject of an existing multicast download session; and

causing the multiple client devices to join an existing multicast group corresponding to the existing multicast download session.

4. (Original) The method of claim 1 wherein applying, by the server, one or more flow control techniques not defined by multicast TFTP comprises modifying quality of service based, at least in part, on resource conditions.

5. (Original) The method of claim 4 wherein modifying the quality of service comprises one or more of: modifying block size and modifying timeout length.

6. (Original) The method of claim 1 wherein applying, by the server, one or more flow control techniques not defined by multicast TFTP comprises reducing a packet transmission rate.

7. (Original) The method of claim 1 wherein applying, by the server, one or more flow control techniques not defined by multicast TFTP comprises retransmitting a most recently transmitted packet in response to receiving an unexpected packet.

8. (Currently Amended) A server device comprising:

a network interface to receive messages from ~~one or more~~ a plurality of client devices including requests to download a file stored by the server device;

a memory coupled with the network interface to store the file; and

a processor coupled with the memory and the network interface, the processor configured to:

execute a multi-threaded application having at least one thread to monitor a port of the server device for received requests from the plurality of client devices;

~~receive a request from a first client device of the one or more client devices to multicast the file as a plurality of packets of data from the server device to the one or more client devices by executing a multi-threaded application with at least one thread monitoring a port designated for receiving multicast requests;~~

execute at least three request handlers to manage any of the received requests from ~~remote~~ the plurality of client devices, the at least three request handlers comprising an upload request handler, a multicast download request handler and a unicast download request handler;

detect that a multicast request is received from a first client device of the plurality of client devices;

invoke the multicast download request handler in order to save session information regarding the received multicast request and create another thread of the multi-threaded application to service the received multicast request;

wherein servicing the received multicast request by the another thread further comprises the processor to:

transmit the plurality of packets of data from **[[**a**]]** the server device to the one or more ~~multiple~~ client devices using a multicast trivial file transfer protocol (TFTP) as a TFTP-compliant flow; and

apply one or more flow control techniques not defined by the multicast TFTP to the TFTP-compliant flow, wherein the flow control techniques comprise**[[**s**]]** ~~at least~~ determining whether the server device has sufficient resources to satisfy the request based on a block size corresponding to the request and an available bandwidth, and sending an error packet to the first client device if the server device does not have sufficient resources to satisfy the request, wherein applying, by the server, one or more flow control techniques not defined by multicast TFTP comprise delaying a start of the transmission of the plurality of packets, wherein delaying includes if a packet is lost and a send delay is determined to be zero, the send delay of zero is set to one, and if the send delay is determined to be greater than timeout/four, the send delay is set equal to the timeout/four, wherein the send delay is doubled or the send delay is

decreased by one for a number of successful packets received until the send delay reaches zero, wherein the number of successful packets received includes ten.

9. (Cancelled)

10. (Original) The server of claim 8 wherein the one or more flow control techniques not defined by multicast TFTP comprises determining whether a request to download the file is a subject of an existing multicast download session, and causing the multiple client devices to join an existing multicast group corresponding to the existing multicast download session.

11. (Original) The server of claim 8 wherein the one or more flow control techniques not defined by multicast TFTP comprises modifying quality of service based, at least in part, on resource conditions.

12. (Original) The server of claim 11 wherein modifying the quality of service comprises one or more of: modifying block size and modifying timeout length.

13. (Original) The server of claim 8 wherein the one or more flow control techniques not defined by multicast TFTP comprises reducing a packet transmission rate.

14. (Currently Amended) A non-transitory computer-readable medium having stored thereon instructions that, when executed by one or more processors, cause the one or more processors to:

execute a multi-threaded application having at least one thread to monitor a port of a server device for received requests from a plurality of client devices;

~~receive a request from a first client device to multicast a file as a plurality of packets of data from a server device to multiple client devices by executing a multi-threaded application with at least one thread monitoring a port designated for receiving multicast requests;~~

execute at least three request handlers to manage any of the received requests from ~~remote~~ the plurality of client devices, the at least three request handlers comprising an upload request handler, a multicast download request handler and a unicast download request handler;

detect that a multicast request is received from a first client device of the plurality of client devices;

invoke the multicast download request handler in order to save session information regarding the received multicast request and create another thread of the multi-threaded application to service the received multicast request;

wherein servicing the received multicast request by the another thread further comprises the server device to:

transmit the plurality of packets of data from **[[**a**]]** the server device to the multiple client devices using a multicast trivial file transfer protocol (TFTP) as a TFTP-compliant flow; and

apply~~, by the server,~~ one or more flow control techniques not defined by the multicast TFTP to the TFTP-compliant flow, wherein the flow control techniques comprise**[[**s**]]** ~~at least~~ determining whether the server device has sufficient resources to satisfy the request based on a block size corresponding to

the request and an available bandwidth, and sending an error packet to the first client device if the server device does not have sufficient resources to satisfy the request, wherein applying, by the server, one or more flow control techniques not defined by multicast TFTP comprise delaying a start of the transmission of the plurality of packets, wherein delaying includes if a packet is lost and a send delay is determined to be zero, the send delay of zero is set to one, and if the send delay is determined to be greater than timeout/four, the send delay is set equal to the timeout/four, wherein the send delay is doubled or the send delay is decreased by one for a number of successful packets received until the send delay reaches zero, wherein the number of successful packets received includes ten.

15. (Cancelled)

16. (Original) The medium of claim 14 wherein the instructions that cause the one or more processors to apply, by the server, one or more flow control techniques not defined by multicast TFTP comprise instructions that, when executed, cause the one or more processors to:

determine whether a request to download the file is a subject of an existing multicast download session; and

cause the multiple client devices to join an existing multicast group corresponding to the existing multicast download session.

17. (Original) The medium of claim 14 wherein the instructions that cause the one or more processors to apply, by the server, one or more flow control techniques not

defined by multicast TFTP comprise instructions that, when executed, cause the one or more processors to modify quality of service based, at least in part, on resource conditions.

18. (Original) The medium of claim 14 wherein the instructions that cause the one or more processors to apply, by the server, one or more flow control techniques not defined by multicast TFTP comprise instructions that, when executed, cause the one or more processors to reduce a packet transmission rate.

Claims 19-22 (Cancelled)

Allowable Subject Matter

Claims 1,3-8,10-14,16-18 are allowed.

The following is an examiner's statement of reasons for allowance:

The provision for --- a method for flow control techniques for TFTP transmission, said method comprising:

invoking, by the server device, the multicast download request handler in order to save session information regarding the received multicast request and create another thread of the multi-threaded application to service the received multicast request;

wherein servicing the received multicast request by the another thread further comprises:

transmitting the plurality of packets of data from the server device to the multiple client devices using a multicast trivial file transfer protocol (TFTP) as a TFTP-compliant flow; and

applying one or more flow control techniques not defined by the multicast TFTP to the TFTP-compliant flow, wherein the flow control techniques comprise determining whether the server device has sufficient resources to satisfy the request based on a block size corresponding to the request and an available bandwidth, and sending an error packet to the first client device if the server device does not have sufficient resources to satisfy the request, wherein applying, by the server, one or more flow control techniques not defined by multicast TFTP comprise delaying a start of the transmission of the plurality of packets, wherein delaying includes if a packet is lost and a send delay is determined to be zero, the send delay of zero is set to one, and if the send delay is determined to be greater than timeout/four, the send delay is set equal to the timeout/four, wherein the send delay is doubled or the send delay is decreased by one for a number of successful packets received until the send delay reaches zero, wherein the number of successful packets received includes ten

--- wherein all the features previously described in the claims are combined in one singular embodiment, is not fairly taught or suggested by the prior art of record.

The Examiner finds particular novelty in the method for TFTP flow control as described in the Applicant Specification (Page 5, Paragraph 13, Page 8, Paragraph 20, and Page 9 Paragraph 23) wherein the said method allows the transfer rate to be controlled by the server device for TFTP transmissions by implementing a multicast download request handler thread (FIG. 5) that saves session information before creating a second thread for TFTP transmission. The method ensures that that the server has sufficient resources to satisfy the request based on a block size corresponding to the request and an available bandwidth, and further implements flow control for said TFTP transmission by implementing a transmission delay. The said transmission delay is calculated by monitoring if a packet is lost and a send delay is determined to be zero, the send delay of zero is set to one, and if the send delay is determined to be greater than timeout/four, the send delay is set equal to the timeout/four, wherein the send delay is doubled or the send delay is decreased by one for a number of successful packets received until the send delay reaches zero, wherein the number of successful packets received includes ten.

Riedle disclosed tracking lost data packets in a multicast TFTP network environment. Upon detection of a last packet (i.e., a packet with less than 512 bytes) or occurrence of a time-out while waiting to receive the next packet, a new master client will re-open the file transfer, request specific packets that are missing, and begin sending ACKs for the packets received until it has received all the packets. Any subsequent clients can start receiving blocks of a file during a transfer and then request

any missing blocks when that client becomes the master client. However Riedle does not disclose a multicast download request handler thread that saves session information before creating a second thread for TFTP transmission. Riedle does not disclose ensuring that that the server has sufficient resources to satisfy the request based on a block size corresponding to the request and an available bandwidth, and further implements flow control for said TFTP transmission by implementing a transmission delay. Riedle does not disclose calculating the transmission delay by monitoring if a packet is lost and a send delay is determined to be zero, the send delay of zero is set to one, and if the send delay is determined to be greater than timeout/four, the send delay is set equal to the timeout/four, wherein the send delay is doubled or the send delay is decreased by one for a number of successful packets received until the send delay reaches zero, wherein the number of successful packets received includes ten.

Paek disclosed using the broadcast TFTP to allow a transmission speed faster than that of an existing one-to-one transmission manner, which makes it possible for multiple clients to download files from one server in a quick and efficient manner. In addition, in the case of adding a block size option item in addition to the broadcast option item extension, the block size larger than a previous block size allows more rapid file transmission. Paek disclosed wherein a server system 50 broadcasts the data in a 1428octet unit only after the T time has elapsed (306), and only after the first client 51a, which is a master client, transmits an acknowledgement message (307). However Paek does not disclose a multicast download request handler thread that saves session

information before creating a second thread for TFTP transmission. However Paek does not disclose ensuring that that the server has sufficient resources to satisfy the request based on a block size corresponding to the request and an available bandwidth, and further implements flow control for said TFTP transmission by implementing a transmission delay. Paek does not disclose calculating the transmission delay by monitoring if a packet is lost and a send delay is determined to be zero, the send delay of zero is set to one, and if the send delay is determined to be greater than timeout/four, the send delay is set equal to the timeout/four, wherein the send delay is doubled or the send delay is decreased by one for a number of successful packets received until the send delay reaches zero, wherein the number of successful packets received includes ten.

Miller1 disclosed determining whether the server device has sufficient resources to satisfy the request based on a block size corresponding to the request and an available bandwidth. Miller1 disclosed a flow control method for data transmission wherein the block sizes are determined along with the transmission rate in order to regulate the transmission flow. Furthermore Miller2 disclosed sending an error packet to the first client device if the server does not have sufficient resources to satisfy the request. However Miller1/Miller2 does not disclose a multicast download request handler thread that saves session information before creating a second thread for TFTP transmission. Miller1/Miller2 does not disclose calculating the transmission delay by monitoring if a packet is lost and a send delay is determined to be zero, the send delay of zero is set to one, and if the send delay is determined to be greater than timeout/four,

the send delay is set equal to the timeout/four, wherein the send delay is doubled or the send delay is decreased by one for a number of successful packets received until the send delay reaches zero, wherein the number of successful packets received includes ten.

Srikantan disclosed monitoring a server port wherein one processor thread is shared among the sockets for detecting events and notifying event consumers. Also, a pool of threads is allocated for executing tasks issued by the various event consumers. Each socket is associated with an event consumer object that is notified when an event is received at the socket. The event consumer objects invoke task objects to handle the events. However Srikantan did not disclose a multicast download request handler thread that saves session information before creating a second thread for TFTP transmission Srikantan does not disclose calculating the transmission delay by monitoring if a packet is lost and a send delay is determined to be zero, the send delay of zero is set to one, and if the send delay is determined to be greater than timeout/four, the send delay is set equal to the timeout/four, wherein the send delay is doubled or the send delay is decreased by one for a number of successful packets received until the send delay reaches zero, wherein the number of successful packets received includes ten.

Xu disclosed detecting when the immediate child computer systems do not acknowledge reception of a sequence of multi-cast packets within a specified period of time (e.g., as a result of network congestion or latency) and decreasing the send rate of multi-cast packets. Accordingly, a root computer system can adjust a send rate to

compensate for changes in the transmission characteristics (e.g., available bandwidth and latency) of networks used to deliver multi-cast packets. However Xu does not disclose a multicast download request handler thread that saves session information before creating a second thread for TFTP transmission. Xu does not disclose calculating the transmission delay by monitoring if a packet is lost and a send delay is determined to be zero, the send delay of zero is set to one, and if the send delay is determined to be greater than timeout/four, the send delay is set equal to the timeout/four, wherein the send delay is doubled or the send delay is decreased by one for a number of successful packets received until the send delay reaches zero, wherein the number of successful packets received includes ten.

Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to GREG BENGZON whose telephone number is (571)272-3944. The examiner can normally be reached on Mon. thru Fri. 8 AM - 4:30 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, William Vaughn can be reached on (571)272-3922. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Greg Bengzon/
Examiner, Art Unit 2444